

# 2023-03-27 Meeting: Continue Date Properties, Activity types/examples and Flattening Examples

---

<a href="#">Charter</a>	<a href="#">Project Page</a>	<a href="#">Terminology</a>	<a href="#">References</a>	<a href="#">Working Documents</a>	<a href="#">Deliverables</a>	<a href="#">Meeting Notes</a>
-------------------------	------------------------------	-----------------------------	----------------------------	-----------------------------------	------------------------------	-------------------------------

---

- [Attendees](#)
  - [Regrets](#)
  - [ACTION ITEMS:](#)
  - [Agenda](#)
  - [Meeting Materials](#)
- 

## Attendees

Dave, Vitus, Steven, Kevin,

## Regrets

Christine, Nancy F,

## ACTION ITEMS:

TODO flesh out scenarios/examples so that we can have any requirements for order and verbosity reflected in our recommendations.

---

## Agenda

- Updates:
  - Steven met with Gloria Gonzalez recently to discuss QA and moving away from caching. Cornell isn't positioned to maintain a cache. ActivityStreams is anticipated to be helpful for maintaining an updated cache.
  - Kevin met with Gloria to describe their AS implementation
    - Currently still using their ATOM feed for production workflows, while the AS implementation settles.
    - Changes may include how the Activities are described, but also workflows for creating the stream to help surface meaningful changes.
    - TODO -Steven will get numbers of label changes we're experiencing via our local parsing strategy.
    - Could you create the cache just using the AS stream? No, because of the away the stream is produced. That's why the data dumps are still important as a starting point. There are now weekly dumps.
    - To access the MARC from id.loc.gov there's a (generous) limit to the HTML and the MARC/XML. This enable short quick batch updates, but not grab the full file.
- Date properties
  - When to use the following in our context (TODO: Steven to inventory Getty's date property usage; Kevin to inventory LC date property usage):
    - [endTime](#) property definition
      - Getty usage - used on each Activity, the value is the same as for the "created" property (see issues with "created" below)
      - LOC usage - not used presently.
    - [startTime](#) property definition
      - Getty usage - not used presently.
      - LOC usage - not used presently.
    - [published](#) property definition
      - Getty usage? - not used presently.
      - LOC usage - used for each Activity Object like "Add" or "Remove" or "Update." I.e., not used for each Object of an Activity. Represents when the Activity Object was 'published.'
    - [updated](#) property definition
      - Getty usage - not used presently.
      - LOC usage - used for each Object of an Activity. Represents the (last) 'update' date of the Object which is the focus of the Activity.
    - [deleted](#) property definition
      - Getty usage - not used presently.
      - LOC usage - not used presently.
    - 'created' - is used by Getty, but understood not to be an AS property; this will be dropped by JSON-LD tooling (ok, because redundant with endTime value)
    - Getty advertises as Create, Update, Delete Types

- LOC advertises as Add, Update, and Remove Types. Deletes is really what they are doing (deprecation is the default to keep URIs from disappear). If changed to Delete, they wouldn't actually delete, still deprecating. Remove is pointing out that these terms are removed from a scheme when deprecated. Add is pointing out that when a term is created, it's added to a scheme.
- Could proceed with these differences, or align if the distinctions aren't important.
  - Moving a term from one scheme to another, not just deleting, is reflected like a normal deprecate with a separate add in another stream.
  - Could have multiple types (Create/Add, Remove/Delete/Change)
    - E.g. "type": ["Create", "Add"],
    - E.g. "type": ["Delete", "Remove", "Update"],
  - From AS, "When an implementation uses an extension type that overlaps with a core vocabulary type, the implementation must also specify the core vocabulary type."
  - Think more about origin/target before deciding on types.
  - TODO does Getty actually delete?
- TODO from last meeting: Dave will create a 7.0, and we can compare it to 7.1 and an eventual 7.2 (for Getty's activity stream). Then we can decide what might be removed and/or added to an appendix
- Continue to discuss examples (updated)
  - Examples: <https://gist.github.com/sfolsom/07338b416ca605d92984c24c346ba5fe>
    - From previous meeting: *Multi typing is probably not the best way to work through the Deprecation, Merging, and Splitting. They may happen over time. Creations of objects that are the new targets needs to happen before Splits and Merges.*
      - *X and Y replaced by Z (full merge), X and Y are deprecated*
      - *Parts X are now considered Z (partial Split)*
      - *Parts X are now considered Z (Partial split), and all of Y are Z (replaced), Y is deprecated*
      - *X is now either Y or Z (full Split), X is deprecated*
      - *What happens in the moment?*
        - *E.g. two names for the same person. You deprecate one and then point to the remaining authority*
        - *E.g. undifferentiated name, some of the things linked to can now be attributed to a specific other name.*
        - *E.g. undifferentiated name, all thinks linked to are now attributed to two or more other names.*
        - *E.g. undifferentiated name, status changed to because everything is attributed to*
        - *E.g. shared pseudonyms TODO flesh out possible changes that occur related to shared pseudonyms.*
        - *E.g. two units within and organization, the two existing units are collapsed into one (the authorities remain), and a new one is used.*
          - *Time period for applicability can be recorded.*

## Meeting Materials

- Recommendations: [https://ld4.github.io/entity\\_metadata\\_management/api/0.1/recommendations.html](https://ld4.github.io/entity_metadata_management/api/0.1/recommendations.html)
  - Examples: [https://ld4.github.io/entity\\_metadata\\_management/api/examples.html](https://ld4.github.io/entity_metadata_management/api/examples.html)
  - Pull requests: [https://github.com/LD4/entity\\_metadata\\_management/pulls](https://github.com/LD4/entity_metadata_management/pulls)
  - Issues: [https://github.com/LD4/entity\\_metadata\\_management/issues](https://github.com/LD4/entity_metadata_management/issues)
  - How to contribute: [https://github.com/LD4/entity\\_metadata\\_management#contributing](https://github.com/LD4/entity_metadata_management#contributing)
- 
-